REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	7/17/96	Final (9/1/94--1/31/96)

4. TITLE AND SUBTITLE

AASERT Fellowship Final Technical Report--Flexible Memory Systems

5. FUNDING NUMBERS

Grant#
F49620-94-1-0462

6. AUTHOR(S)

Nicholas Carter and William J. Dally

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

AFOSR-TR
96-0410

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

nm

Air Force Office of Scientific Research
110 Duncan Avenue, Suite B115
Bolling Air Force Base, D.C. 20332-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

19960805 031

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unlimited

12. ABSTRACT (Maximum 200 words)

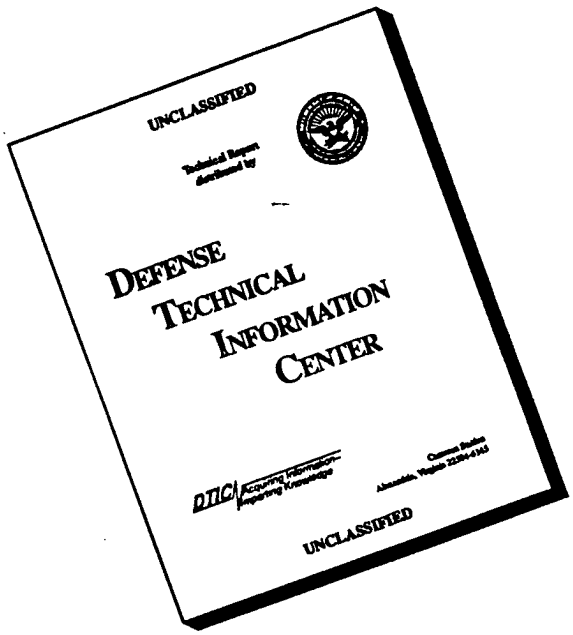
This is the final report on the work done under the Flexible Memory Systems AASERT fellowship (Grant #F49620-94-1-0462). During the funding period, progress was made in two major areas:

- M-Machine Hardware: The memory system architecture of the M-Machine was finalized. RTL (Verilog) implementation of the memory system was continued. (RTL implementation began before the start of the funding period) At the end of the funding period, RTL models of the M-Machine's cache and external memory interface had been developed. The external memory interface had been integrated with a model for the M-Machine's off-chip memory, and integration of the cache banks and external memory interface had begun.
- Runtime Software: System software for the M-Machine was developed which use the novel features of the M-Machine to implement shared memory in software with hardware assistance, demonstrating that the M-Machine's memory mechanisms are sufficient for implementation of shared memory.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Memory Systems, Cache Coherence	3
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AASERT Fellowship Final Technical Report -- Flexible Memory Systems

Nicholas Carter and William J. Dally

July 17, 1996

Abstract

This is the final report on the work done under the Flexible Memory Systems AASERT fellowship (Grant #F49620-94-1-0462). During the funding period, progress was made in two major areas:

- M-Machine Hardware: The memory system architecture of the M-Machine was finalized. RTL (Verilog) implementation of the memory system was continued. (RTL implementation began before the start of the funding period) At the end of the funding period, RTL models of the M-Machine's cache and external memory interface had been developed. The external memory interface had been integrated with a model for the M-Machine's off-chip memory, and integration of the cache banks and external memory interface had begun.
- Runtime Software: System software for the M-Machine was developed which use the novel features of the M-Machine to implement shared memory in software with hardware assistance, demonstrating that the M-Machine's memory mechanisms are sufficient for implementation of shared memory.

1 Introduction

This document serves as the final technical report on the work funded by the Flexible Memory Systems AASERT award. The goals of this work were to extend the scope of the M-Machine project to provide greater flexibility in the memory system of the M-Machine, as part of his ongoing work as the memory system architect of the M-Machine.

The memory system of the M-Machine takes a somewhat unusual approach to implementing shared memory, by providing hardware support for shared memory without restricting the machine to a particular shared memory protocol. Previous machines have tended to either provide hardware to support a particular shared memory model, which allows the machine to implement that model efficiently, but limits the machine to one shared memory model, or to provide no hardware to support shared memory, which allows flexibility, but tends to result in poor performance due to software overheads.

The memory system of the M-Machine is designed to provide both efficiency and flexibility in shared memory. Instead of providing hardware to support one specific shared memory model, the M-Machine provides hardware mechanisms which accelerate a number of functions which are common to many shared memory implementations. These hardware mechanisms allow the M-Machine's hardware to determine which references can be resolved locally, and which access remote data. Local memory references are handled quickly in hardware, while remote references require software intervention. This model allows the efficient implementation of a number of different shared memory models, as the hardware mechanisms reduce the frequency with which software is invoked, and the run-time of the handlers when they are invoked.

2 Progress and Accomplishments

The AASERT award covered work on the memory system of the M-Machine as part of the M-Machine project, which continues to this day. During the period covered by the award, progress was made in the areas of architectural refinement, RTL Implementation, and Run-Time Software.

2.1 Architectural Refinement

During the period covered, a number of refinements were made to the architecture of the M-Machine's memory system. Due to the fact that the basic architecture of the M-Machine was fairly well-defined, the architectural refinements that were made were minor refinements as opposed to major changes. Refinements to the M-Machine architecture during this time period:

- The external memory system was made responsible for generating all event queue entries for events raised by the memory system. If the cache banks detect an event, the request is forwarded to the EMI for handling. Previously, the architecture had called for the cache banks to generate event queue entries for events that they detect. This architecture change was made to simplify the cache banks.
- The event queue format was changed to improve the performance of the runtime software by having the hardware generate data that the runtime software was having to compute, and including both the physical and virtual addresses of a memory event in the event queue entry to eliminate the need for the event handlers to perform an address translation.
- The PUTCSTAT instruction was revised so that it atomically flushes a data block from the on-chip cache, changes the block status of that block to the specified value, and returns the previous block status of the block. This change was made because the runtime software needs the ability to determine the status of a cache block immediately before the block is invalidated in the cache, and providing an atomic instruction to do this was the only way to provide this function.

2.2 RTL Implementation

The RTL implementation of the M-Machine continued throughout the funding period. During the funding period, the RTL for the cache banks and the EMI were completed to a sufficient level that integration testing of the entire memory system could begin. By the end of the funding period, the integrated memory system could perform reads and writes to both the caches and the external memory, and could detect and handle events by writing event queue entries into the event queue. The schematic design for the cache datapath was begun during the funded period, and completed shortly after the funded period.

2.3 Run-time Software

During the funded period, work was begun on implementing the run-time system of the M-Machine, including software handlers to allow the execution of shared-memory programs on the M-Machine. Two remote memory handlers have been implemented on the M-Machine, one which implements a cached shared-memory scheme, and one which implements an uncached shared-memory scheme. The implementation of these memory handlers serves to drive the refinement of the memory system architecture, and to demonstrate that the M-Machine's memory system mechanisms can be used to implement shared memory successfully.

3 Publications

Two major publications were made based on the work covered by this award. In 10/94, the paper "Hardware Support for Fast Capability-Based Addressing," which describes the memory protection scheme of the M-Machine, was published in the proceedings of the 6th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS VI). In 11/95, the paper "The M-Machine Multicomputer," by Marco Fillo *et. al.* was published in the MICRO-28 conference. This paper described the architecture of the M-Machine, including a description of the memory system and preliminary results from the run-time system. Copies of both of these papers are included with this report.

4 Conclusion

During the period covered by this award, a number of refinements were made to the memory system architecture, significant progress was made on the implementation of the M-Machine hardware, and run-time software was written to demonstrate the feasibility of the M-Machine's memory system as a platform for implementing shared memory.

GRANT NUMBER:

FORM A2-2

AUGMENTATION AWARDS FOR SCIENCE & ENGINEERING RESEARCH TRAINING (AASERT)
REPORTING FORM

The Department of Defense (DOD) requires certain information to evaluate the effectiveness of the AASERT program. By accepting this Grant Modification, which bestows the AASERT funds, the Grantee agrees to provide the information requested below to the Government's technical point of contact by each annual anniversary of the AASERT award date.

1. Grantee identification data: (R & T and Grant numbers found on Page 1 of Grant)

a. Massachusetts Institute of Technology
University Name

b. F49620-94-1-0462
Grant Number

c. _____
R & T Number

d. William J. Dally
P.I. Name

e. From: 3/1/95 To: 2/29/96
AASERT Reporting Period

NOTE: Grant to which AASERT award is attached is referred to hereafter as "Parent Agreement."

2. Total funding of the Parent Agreement and the number of full-time equivalent graduate students (FTEGS) supported by the Parent Agreement during the 12-month period prior to the AASERT award date.

a. Funding: $ 877,494

b. Number FTEGS: _____

3. Total funding of the Parent Agreement and the number of FTEGS supported by the Parent Agreement during the current 12-month reporting period.

a. Funding: $ 94,553

b. Number FTEGS: 4.8

4. Total AASERT funding and the number of FTEGS and undergraduate students (UGS) supported by AASERT funds during the current 12-month reporting period.

a. Funding: $ 78,753

b. Number FTEGS: 2

c. Number UGS: _____

VERIFICATION STATEMENT: I hereby verify that all students supported by the AASERT award are U.S. citizens.

Principal Investigator

7/18/96
Date